

Remarks/Arguments

Claims 1-43 are pending in the application. Claims 1, 21 and 42 are independent.

Claims 1-43 were rejected.

Claim 43 was objected to for antecedent basis.

Claims 2, 4, 5, 27, 24 and 25 are cancelled.

Claims 1, 6, 9, 12, 15, 17, 19, 21, 29, 32, 35, 37, 39, 42 and 43 are amended.

Claim 43 is amended herein to rectify the antecedent basis problem noted. More specifically, claim 43 is amended to clarify and state more precisely that the interface component is obtained by using the keywords included in the access request to search the data structure.

Claims 9, 12, 15, 17, 18, 19, 29, 32, 35, 37, 39 and 39 are amended to be definite and complete with regard to the Markush groups respectively claimed therein.

Claim 1 is amended to incorporate the subject matter of former claims 2 and 5. Claims 21, 42 and 43 are similarly amended.

No new matter has been added by way of these claim amendments.

Claim Rejections Under 35 USC 112

Claims 9-20 and 29-41 were rejected under 35 USC 112, second paragraph.

Claims 9, 12, 15, 17, 18, 19, 29, 32, 35, 37, 38 and 39 have been amended to be definite and complete as to the membership of the Markush groups claimed respectively therein.

Claim Rejections Under 35 USC 102

Claim 1 as amended recites: *a method for providing dynamic interaction between a pair of application programs by an interface module of a terminal, the pair of applications including a requestor application desiring access to a target application, the method comprising the steps of:*

registering access information of the target application, the access information including published access information made available in a data structure for retrieval by the interface module;

receiving an access request by the interface module from the requestor application, the access request including content corresponding to the published access information of the target application;

obtaining an interface component by using the content to search the data structure, the interface component including an application program interface (API) configured in a language incompatible with the interface module, the interface component further including an access handler for translating the incompatible language between the API and the interface module; and

employing the interface component by the interface module to satisfy the access request of the requestor application for interaction with the target application.

The Examiner has rejected claims 1-7, 12-14, 19-27, 32-34 and 39-42 under 35 U.S.C. 102(e) as being anticipated by Slaughter, US patent no. 7,458,082. Applicant respectfully traverses the rejections.

As described in the Background of the application as filed, the problem the subject invention addresses relates to communication between related applications. Specifically, if an application interface changes, it is also required

to change many, or all, of the related or dependent applications to maintain compatibility and interoperability.

The claim 1 element of *"...the interface component including an application program interface (API) configured in a language incompatible with the interface module, the interface component further including an access handler for translating the incompatible language between the API and the interface module.."*, as claimed in former claims 2 and 5 (now cancelled) of the present application is not disclosed in Slaughter.

In rejecting the former claim 5 (the subject matter of which is not incorporated into claim 1 via the amendment herein) the Examiner cited col. 7 lines 50- 60 of Slaughter as disclosing *"wherein the interface component further includes an access handler configured for providing translation between the interface module and the application program interface"*:

One embodiment of a distributed computing environment may include service proxies. A service proxy is a service that implements the distributed computing environment protocol on behalf of a foreign service. Service providers in the distributed computing environment may respond to the distributed computing environment protocol suite. For services that do not support the distributed computing environment, a service proxy may translate messages to and from the proxied service's native invocation model.

55

It is apparent, however, that this cited passage of Slaughter is simply a proxy that converts the native application into a distributed computing model protocol, so the application does not need to know the underlying protocol details.

In contrast, the translation process of the claimed subject matter of claim 1 using the access handlers allow the calling application to access any API, whether native or otherwise.

Therefore, for at least the reasons discussed above, Applicant submits claim 1 as amended herein is patentable in view of Slaughter and, as such, requests that the rejection of claim 1 be withdrawn.

Independent claims 21 and 42 include similar limitations as claim 1, and therefore a corresponding argument applies. Accordingly, Applicant submits that the rejection to these claims be withdrawn for at least the same reasons discussed above with regard to the Slaughter reference.

Since the remaining dependent claims depend from one of the above noted independent claims, Applicant submits that the rejection of these claims be withdrawn for at least the same reasons, since none of the cited Slaughter, Loo or Bloch references disclose the claim element discussed above.

For the foregoing reasons, the Applicant respectfully submits that the claimed invention is patentable over the prior art. Reconsideration and allowance are respectfully requested.

Respectfully submitted,

/Henry Ohab/

Henry Ohab
Registration No. 45,854
416 862 3593

Gowling Lafleur Henderson LLP
1 First Canadian Place, Suite 1600
Toronto, Ontario, M5X 1G5
TOR_LAW\ 73570511